

Direct Polarization for q-ary Source and Channel Coding

Ángel Bravo-Santos*, *Senior Member, IEEE*,

Abstract—It has been shown that an extension of the basic binary polar transformation also polarizes over finite fields. With it the direct encoding of q-ary sources and channels is a process that can be implemented with simple and efficient algorithms. However, direct polar decoding of q-ary sources and channels is more involved. In this paper we obtain a recursive equation for the likelihood ratio expressed as a LR vector. With it successive cancellation (SC) decoding is applied in a straightforward way. The complexity is quadratic in the order of the field, but the use of the LR vector introduces factors that soften that complexity. We also show that operations can be parallelized in the decoder. The Bhattacharyya parameters are expressed as a function of the LR vectors, as in the binary case, simplifying the construction of the codes. We have applied direct polar coding to several sources and channels and we have compared it with other multilevel strategies. The direct q-ary polar coding is closer to the theoretical limit than other techniques when the alphabet size is large. Our results suggest that direct q-ary polar coding could be used in real scenarios.

Index Terms—Source coding, channel coding, Gaussian channels, quadrature amplitude modulation.

I. INTRODUCTION

The basic polar transformation for channel coding presented in [1] maintains the polar properties for certain alphabets and channels. The authors in [2] prove that, under certain conditions, the mutual information of q-ary channels polarizes. Moreover, if the input alphabet is of prime size with a modulo addition defined in it, the basic transformation polarizes. In the same paper it is also shown that for some non-prime alphabets it is possible to find channels for which the basic transform does not polarize.

In [3] and [4] it is shown that for alphabets whose size is a power of a prime the basic polar transformation for channel coding polarizes with more than two levels. Specific definitions of Bhattacharyya parameters allow the construction of information sets that, eventually, allow for reliable transmission of information. Further, the multilevel channel polar coding reaches the symmetric capacity.

Polar codes have been applied to continuous channels using some form of multilevel modulation. In [5] generalized concatenation codes implemented using polar codes are applied to pulse amplitude modulation (PAM). In [6] a multilevel code is defined as a binary partition of the multilevel channel that

is concatenated with a polar code. It is also shown in [6] that this scheme reaches the capacity of the symmetric channel.

Binary polar codes have been used for lossy source coding using distortion criteria [7]. In [8] lossless source coding was described based on the dual channel approach. A more direct approach to source coding with binary polar codes was provided by Arikan in [9], where it was also shown that channel coding can be interpreted as an instance of source coding.

The extension of binary polar codes for source coding to the q-ary case under a distortion criterion was presented in [10]. The work [11] generalizes lossless polar source coding to arbitrary alphabets using the Arikan's direct source coding approach [9]. In [11] it was also proved that the basic polar transform [1] always polarize with alphabets of prime size, but there are alphabets of non-prime size where it is possible to find channels where the polarization is not possible. In the same paper the channel coding problem is treated as an instance of source coding. In [12] it is shown that the basic transformation can be generalized allowing polarization of sources and channels in finite fields.

There is a well known optimal symbol-by-symbol decoding rule in the literature [13] for linear codes with symbols defined on a finite field. Its complexity is exponential in the dimension of the dual code and therefore more efficient alternatives are used. The usual way of implementing “good” codes with q-ary alphabets, or q-ary signal sets, is by means of constructing first “good” binary codes, where “good” refers to the proximity to the theoretical limit [14]. The main objective of this work is to find the theoretical tools to implement and construct polar codes that work natively with q-ary alphabets, or q-ary signal sets. Therefore this direct approach, that works with symbols from a finite field and with a polar transformation also defined in that field [15], avoids the use of binary codes. In previous works [11], [12], no explicit expression or algorithm was provided for decoding.

In this paper we use the LR vector of the encoded symbols given the side information for source coding, or the LR vector of the unencoded symbols given the channel observations, for non-binary polar codes. We then derive a recursive expression for the LR vector that allows to perform successive cancellation (SC) decoding. The Bhattacharyya parameters are obtained as a function of the LR. As a result, the problems of q-ary source and channel coding, and code construction, are solved as in [1] with complexity of the same order as in the binary case in the encoder and about $q^2/2$ times the binary complexity in the decoder.

In this paper we also perform a comparison of q-ary polar

This work has been partially supported by the Spanish government (TEC2012-38883-C02-01, Consolider-Ingenio 2010 CSD2008-00010).

A. Bravo-Santos is with the Departamento de Teoría de la Señal y Comunicaciones, Universidad Carlos III de Madrid, Avda. de la Universidad 30, Leganés, 28911 Madrid, Spain (abravo@tsc.uc3m.es), Tel: +34 91 624 8752, Fax: +34 91 624 8749.

coding for channel coding with multilevel coding (MLC) and bit-interleaved coded modulation (BICM) using simulations. The results suggest that q-ary polar codes are closer to the theoretical limit that other codes as the alphabet size is large. There is not need to optimize the labeling symbols-signals in channel coding with q-ary polar codes. However, this is not the case of other multilevel coding alternatives. Furthermore, the relation between channel degradation and Bhattacharyya parameters of the input symbols in q-ary polar coding is similar to the binary case, as it is shown in the paper. From this fact it is straight forward to extend the benefits of binary polar coding to the q-ary case, such as, rate adaptation [16], network coding [17], [18], etc. This extension is not so immediate with MLC or BICM. The complexity of q-ary polar decoding is higher than that of MLC or BICM. However, as it is discussed in the paper, many operations can be done in parallel, as opposed to the serial definition of MLC.

In this work we follow the notation in [1] and we do not provide additional explanation about it. The rest of the paper is organized as follows. In Section II we formulate the problem. In Section III we present the recursive LR equations and Bhattacharyya parameter as a function of the LR. In Section IV we analyze the implementation of the source and channel polar coding. In Section V we describe the simulations conducted with q-ary sources and multilevel polar coding on the additive white Gaussian noise (AWGN) channel. The main conclusions about the results of this work are in Section VI.

II. SOURCE CODING WITH POLAR CODES OVER FINITE FIELDS

We consider a discrete memoryless source with alphabet \mathcal{X} formed by the elements of the Galois field \mathbb{F}_q . Associated to a source symbol X is the side information $Y \in \mathcal{Y}$. We have $(X, Y) \in \mathcal{X} \times \mathcal{Y}$. The pair (X, Y) can be view as a memoryless source, $(X, Y) \sim P_{X,Y}(X, Y)$, where $P_{X,Y}(X, Y)$ is the joint distribution of (X, Y) [9]. By taking symbols indepdently from this source we form the sequence

$$(X_1^N, Y_1^N) = ((X_1, Y_1), (X_2, Y_2) \cdots (X_N, Y_N)), \quad N = 2^n, \quad n \in \mathbb{N}.$$

We compress the source sequence X_1^N using a polar code. The basic polar transformation was introduced in [1] and extended in [12] for \mathbb{F}_q . The extended transformation is,

$$\begin{aligned} U_1 &= X_1 + \alpha X_2 \\ U_2 &= X_2, \end{aligned} \quad (1)$$

where $U_1, U_2, X_1, X_2 \in \mathbb{F}_q$, the addition is in \mathbb{F}_q , and α is a constant that is a primitive element of the field if \mathbb{F}_q is a non-prime field. If the field is prime then $\alpha = 1$. The transformation (1) polarizes over \mathbb{F}_q , $q = p^m$, $m, p \in \mathbb{N}$, $m > 1$, p prime [12], or over \mathbb{F}_p [2], [11]. Based on (1) the encoder for the sequence (X_1^N, Y_1^N) follows the classical scheme for polar channel coding described in [1]. In Fig. 1 we show the structure implemented in the encoder. It is easy to expresses the transformation of Fig. 1 as

$$U_1^N = X_1^N G_N, \quad (2)$$

where G_N is the matrix of the transformation.

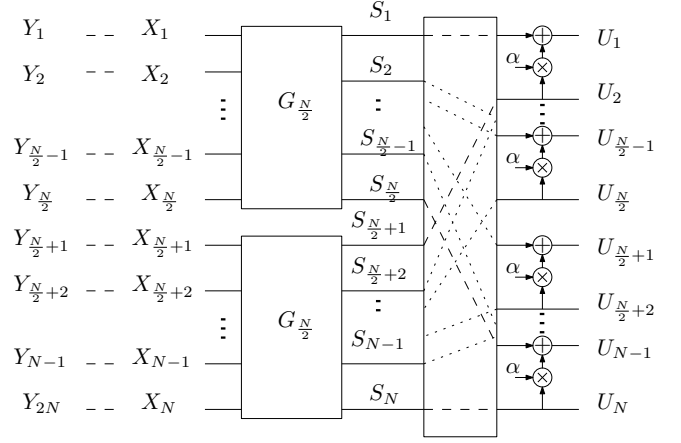


Fig. 1. q-ary polar encoder.

In [19] and [12] it is proved that the entropy of U_i given (Y_1^N, U_1^{i-1}) polarizes on \mathbb{F}_p with the transformation (1). A similar result was proved in [12] for \mathbb{F}_q . For $\epsilon > 0$

$$\begin{aligned} \lim_{N \rightarrow \infty} \frac{1}{N} |\{i : H(U_i | Y_1^N, U_1^{i-1}) > 1 - \epsilon\}| &= H(X|Y), \\ \lim_{N \rightarrow \infty} \frac{1}{N} |\{i : H(U_i | Y_1^N, U_1^{i-1}) < \epsilon\}| &= 1 - H(X|Y), \end{aligned} \quad (3)$$

where the logarithms are to base q . A conclusion from (3) is that if N is large enough we only have two possibilities related to the amount of information about U_i given (Y_1^N, U_1^{i-1}) : a) if $H(U_i | Y_1^N, U_1^{i-1}) \approx 0$ the knowledge of (Y_1^N, U_1^{i-1}) is all we need for recovering U_i ; b) if $H(U_i | Y_1^N, U_1^{i-1}) \approx 1$, the pair (Y_1^N, U_1^{i-1}) does not give information about U_i .

In [9] Arikan gave a definition of the Bhattacharyya parameter of a binary source. This definition was extended in [19] for non-binary alphabets. For a generic source (X, Y) , $X \in \mathcal{X}$, the Bhattacharyya parameter is

$$Z(X|Y) = \frac{1}{|\mathcal{X}| - 1} \sum_{\substack{x, x' \in \mathcal{X} \\ x \neq x'}} \sum_y \sqrt{P_{X,Y}(x, y) P_{X,Y}(x', y)}. \quad (4)$$

For the variables $(U_i; Y_1^N, U_1^{i-1})$, $U_i \in \mathbb{F}_q$, we have,

$$\begin{aligned} Z(U_i | Y_1^N, U_1^{i-1}) &= \frac{1}{q - 1} \\ &\times \sum_{\substack{u_i, u'_i \in \mathbb{F}_q \\ u_i \neq u'_i}} \sum_{y_1^N, u_1^{i-1}} \sqrt{P_N^{(i)}(u_i, y_1^N, u_1^{i-1}) P_N^{(i)}(u'_i, y_1^N, u_1^{i-1})} \\ &= \frac{1}{q - 1} \sum_{y_1^N, u_1^{i-1}} P(y_1^N, u_1^{i-1}) \\ &\times \sum_{\substack{u_i, u'_i \in \mathbb{F}_q \\ u_i \neq u'_i}} \sqrt{P_N^{(i)}(u_i | y_1^N, u_1^{i-1}) P_N^{(i)}(u'_i | y_1^N, u_1^{i-1})}, \end{aligned} \quad (5)$$

where the superscript (i) in $P_N^{(i)}$ has been introduced to establish a parallelism between $P_N^{(i)}$ and the coordinate channel $W_N^{(i)}$ in [1].

The parameters $Z(U_i | Y_1^N, U_1^{i-1})$ and $H(U_i | Y_1^N, U_1^{i-1})$, $i = 1, \dots, N$, polarize simultaneously as it is shown in [9]

for the binary case, in [19] for a p-ary source, and in [12] for a q-ary source. An upper bound to the rate of polarization was obtained in [20] for binary channels with a polarizing transformation defined by a matrix G . The bound was defined with the help of the partial distances defined in the same paper. In [19] the bound to the rate of polarization was extended for sources with prime alphabets and in [12] for sources with alphabet \mathbb{F}_q . The basic polar transformation [9] and the extended transformation in (1) have associated matrices with the same partial distances. Therefore they have the same bound for the rate of polarization. From [19] the rate of polarization for any β , $0 < \beta < 1/2$ is,

$$\lim_{N \rightarrow \infty} \frac{1}{N} |\{i : Z(U_i | Y_1^N, U_1^{i-1}) \leq 2^{-N^\beta}\}| = 1 - H(X|Y) \quad (6)$$

III. DECOMPRESSION VIA RECURSIVE EVALUATION OF THE LR

In this Section we present a novel recursive expression for the LR vector of the encoded symbols given the side information that allows an effective implementation of the decoder. We also express the Bhattacharyya parameter of U_i given (Y_1^N, U_1^{i-1}) as a function of the LR. With the Bhattacharyya parameters the construction of the q-ary polar codes is done in an effective way using Monte Carlo. The LR vector is defined in a similar way as in [21]. For its construction we choose the additive identity of \mathbb{F}_q , the element “0”, as a reference and perform the ratio of the probabilities of the remaining elements to the probability of the reference one. As a consequence, the length of the likelihood ratio vector is $q - 1$. The components of the LR vectors $\mathbf{L}_N^{(i)}(\cdot | y_1^N, u_1^{i-1})$, $i = 1 \dots N$, are,

$$L_N^{(i)}(u | y_1^N, u_1^{i-1}) = \frac{P_N^{(i)}(u | y_1^N, u_1^{i-1})}{P_N^{(i)}(0 | y_1^N, u_1^{i-1})}; \quad u \in \mathbb{F}_q, \quad u \neq 0. \quad (7)$$

Each component of $\mathbf{L}_N^{(i)}(\cdot | y_1^N, u_1^{i-1})$ is associated with a non null element of \mathbb{F}_q .

Claim 1: The LR vectors can be obtained recursively,

$$\begin{aligned} L_N^{(2i-1)}(u | y_1^N, u_1^{2i-2}) &= \\ \frac{\sum_{u_{2i}} L_{\frac{N}{2}}^{(i)}(u - \alpha u_{2i} | y_1^{\frac{N}{2}}, u_{1,o}^{2i-2} - \alpha u_{1,e}^{2i-2}) L_{\frac{N}{2}}^{(i)}(u_{2i} | y_{\frac{N}{2}+1}^N, u_{1,e}^{2i-2})}{\sum_{u_{2i}} L_{\frac{N}{2}}^{(i)}(-\alpha u_{2i} | y_1^{\frac{N}{2}}, u_{1,o}^{2i-2} - \alpha u_{1,e}^{2i-2}) L_{\frac{N}{2}}^{(i)}(u_{2i} | y_{\frac{N}{2}+1}^N, u_{1,e}^{2i-2})}, \\ L_N^{(2i)}(u | y_1^N, u_1^{2i-2}, u_{2i-1}) &= \\ \frac{L_{\frac{N}{2}}^{(i)}(u_{2i-1} - \alpha u | y_1^{\frac{N}{2}}, u_{1,o}^{2i-2} - \alpha u_{1,e}^{2i-2})}{L_{\frac{N}{2}}^{(i)}(u_{2i-1} | y_1^{\frac{N}{2}}, u_{1,o}^{2i-2} - \alpha u_{1,e}^{2i-2})} L_{\frac{N}{2}}^{(i)}(u | y_{\frac{N}{2}+1}^N, u_{1,e}^{2i-2}), \end{aligned} \quad (8)$$

for $u \in \mathbb{F}_q, u \neq 0$, where $i = 1 \dots \frac{N}{2}$, and a negative index of a conditioning variable represents that there is not dependence on this variable. The recursion starts with $N = 1$.

Proof: See Appendix A.

From the LR vector the detection is performed in two steps:

a) the component of $\mathbf{L}_N^{(i)}(\cdot | y_1^N, u_1^{i-1})$ with maximal value is obtained; b) if this component is larger than 1 the associated symbol is output. Otherwise the symbol “0” is output.

A. Bhattacharyya parameter

Binary polar codes can be constructed from the LR in the decoding process [1]. The similarities between the LRs in [1] and the recursive LRs in (8) suggest that the Bhattacharyya parameters can be expressed as a function of the LRs. To this end we multiply and divide the second term of (5) by $P_N^{(i)}(\beta | y_1^N, u_1^{i-1})$, where $\beta \in \mathbb{F}_q$,

$$\begin{aligned} Z(U_i | Y_1^N, U_1^{i-1}) &= \frac{1}{q-1} \sum_{y_1^N, u_1^{i-1}} P(y_1^N, u_1^{i-1}) P_N^{(i)}(\beta | y_1^N, u_1^{i-1}) \\ &\times \sum_{\substack{u_i, u'_i \in \mathbb{F}_q \\ u_i \neq u'_i}} \sqrt{\frac{P_N^{(i)}(u_i | y_1^N, u_1^{i-1}) P_N^{(i)}(u'_i | y_1^N, u_1^{i-1})}{P_N^{(i)}(\beta | y_1^N, u_1^{i-1}) P_N^{(i)}(\beta | y_1^N, u_1^{i-1})}} \end{aligned} \quad (9)$$

After multiplying and dividing the second term of (9) by $P_N^{(i)}(0 | y_1^N, u_1^{i-1})$ we have,

$$\begin{aligned} Z(U_i | Y_1^N, U_1^{i-1}) &= \frac{1}{q-1} \sum_{y_1^N, u_1^{i-1}} P(y_1^N, u_1^{i-1}) P_N^{(i)}(\beta | y_1^N, u_1^{i-1}) \frac{1}{L_N^{(i)}(\beta | y_1^N, u_1^{i-1})} \\ &\times \underbrace{\sum_{\substack{u_i, u'_i \in \mathbb{F}_q \\ u_i \neq u'_i}} \sqrt{L_N^{(i)}(u_i | y_1^N, u_1^{i-1}) L_N^{(i)}(u'_i | y_1^N, u_1^{i-1})}}_{(p-1)\Delta(y_1^N, u_1^{i-1})} \\ &= \sum_{y_1^N, u_1^{i-1}} P(y_1^N, u_1^{i-1}, u_i = \beta) \frac{\Delta(y_1^N, u_1^{i-1})}{L_N^{(i)}(\beta | y_1^N, u_1^{i-1})} \\ &= \sum_{y_1^N, u_1^N, u_i = \beta} P(y_1^N, u_1^N) \frac{\Delta(y_1^N, u_1^{i-1})}{L_N^{(i)}(\beta | y_1^N, u_1^{i-1})}, \end{aligned} \quad (10)$$

where $\Delta(y_1^N, u_1^{i-1})$ is defined as indicated in (10).

From (10) the Bhattacharyya parameters are calculated using Monte Carlo from the LR vector used for decoding as in polar coding for the binary channel [1]. In (10) $Z(U_i | Y_1^N, U_1^{i-1})$ has been obtained for a fixed value $u_i = \beta$. In the numerical simulations we estimate $Z(U_i | Y_1^N, U_1^{i-1})$ by averaging the results obtained over all possible values of u_i . In [1] the construction of binary polar codes is based on the Bhattacharyya parameters. From (10) the construction of q-ary polar codes can be carried out in an effective way.

In the next Section we discuss the construction and implementation of q-ary polar codes for sources and channels.

IV. CODING WITH Q-ARY POLAR CODES

The polar transformation presented in (1) for two variables, and extended in Fig. 1 for N variables, can be used for source coding with side information, or for channel coding [9]. Though the problem is very similar in both cases, there are, however, subtle differences.

A. Source Coding Implementation and Code Construction

The information to be compressed is X_1^N and Y_1^N is the side information available for decoding. Let \mathcal{A} be the set of

indices with low Bhattacharyya parameters,

$$\mathcal{A} = \{i : Z(U_i|Y_1^N, U_1^{i-1}) < \delta\}, \quad (11)$$

with $|\mathcal{A}^c| = \lceil R_s N \rceil$, $R_s > H(X|Y)$. R_s is the code rate for source compression and $\lceil \cdot \rceil$ is the ceil function of its argument. The set \mathcal{A} is called information set in [1] and the set \mathcal{A}^c is called high entropy index set [9], as it gathers the variables with high conditional entropy. The variables $u_{\mathcal{A}}$ have low conditional entropy. They can be recovered using SC decoding with (8). However, the variables $u_{\mathcal{A}^c}$, the frozen variables, have high entropy and are not recoverable with SC decoding. Therefore, $u_{\mathcal{A}^c}$ is the compressed pattern of X_1^N .

The detection is done in two steps: 1) recovering of U_1^N from $u_{\mathcal{A}^c}$ and Y_1^N , 2) obtaining X_1^N from U_1^N .

1) *Recovering U_1^N* : We have to detect the components of U_1^N that are in the information set \mathcal{A} . The rest, $U_{\mathcal{A}^c}$, are available. As we already know $Y_1^N = y_1^N$, we can obtain $P(x_i|y_i)$, $x_i \in \mathbb{F}_q$. Therefore, for each coordinate y_i we can calculate the LR vector $L_1^{(i)}(\cdot|y_i)$ with components

$$L_1^{(i)}(u|y_i) = \frac{P(u|y_i)}{P(0|y_i)}; \quad u \in \mathbb{F}_q, \quad u \neq 0. \quad (12)$$

After the initial step (12) the recursion (8) is repeated $\log_2 N$ times. The LR vectors with indices in \mathcal{A} are detected as indicated in Section III.

2) *Obtaining X_1^N* : The application of the inverse transform G_N^{-1} to U_1^N gives X_1^N , i.e.,

$$X_1^N = U_1^N G_N^{-1}. \quad (13)$$

For a binary polar code $G_N = G_N^{-1}$. However, this is not true for a general q-ary polar code.

The information set \mathcal{A} allows us to upper bound the error probability. If the detected pattern is \hat{X}_1^N , the error probability is,

$$P_e = \Pr(\hat{X}_1^N \neq X_1^N). \quad (14)$$

From [1] and [19] we can show that the error probability is bounded by the Bhattacharyya parameters associated to the information set,

$$P_e \leq (q-1) \sum_{i \in \mathcal{A}} Z(U_i|Y_1^N, U_1^{i-1}). \quad (15)$$

Thus, the sum $\sum_{i \in \mathcal{A}} Z(U_i|Y_1^N, U_1^{i-1})$ is an important parameter for the code construction.

The polar source coding reaches the theoretical limit, $R_s \rightarrow H(X|Y)$ as $N \rightarrow \infty$, as it can be seen from (6) and (11).

B. Channel Coding

The problem of channel coding can be viewed as an instance of source coding where the side information Y_1^N is the channel output. The input to the channel X_1^N can be understood as the information to be compressed in the source coding problem. Therefore, the information to be transmitted to the channel S_1^K must have a similar role as the symbols that can be recovered with SC decoding in the source coding case, $U_{\mathcal{A}} = S_1^K$. In the source coding problem the high entropy symbols $u_{\mathcal{A}^c}$ are available to both the encoder and the decoder. In channel coding this is not possible, in general.

One way to overcome this is to generate the frozen symbols with synchronized pseudorandom generators on the transmitter and receiver sides of the channel. X_1^N is obtained through an inverse polar transform G_N^{-1} of the combined vector formed from the message and frozen symbols. Fig. 2 depicts the scheme of the channel polar coding system, where the block Π is a permutation that produces U_1^N from the message and the frozen symbols.

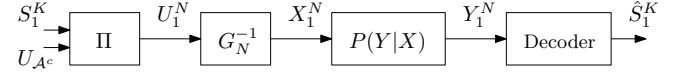


Fig. 2. Channel coding.

The code rate for channel coding is $R_c = |\mathcal{A}|/N$ and, if the message symbols are independent and identically distributed (iid) then, as $N \rightarrow \infty$ [19]

$$R_c \rightarrow 1 - H(X|Y), \quad (16)$$

where the log in the entropy is base q . The q -ary polar code can reach the symmetric capacity, but not the real capacity; however, by increasing the source alphabet it is possible to apply some procedures to reach it [19].

A relation between channel degradation and the Bhattacharyya parameters of binary channels was presented in [7]. The definition of channel degradation [7] is.

Definition 2: Let $P^{(1)}(y_1|x) : \mathbb{F}_q \rightarrow \mathcal{Y}_1$ and $P^{(2)}(y_2|x) : \mathbb{F}_q \rightarrow \mathcal{Y}_2$ two q -ary discrete memoryless channels (DMC). $P^{(1)}$ is degraded with respect to $P^{(2)}$, $P^{(1)} \preceq P^{(2)}$, if there is a channel $P(y_1|y_2) : \mathcal{Y}_2 \rightarrow \mathcal{Y}_1$ such that,

$$P^{(1)}(y_1|x) = \sum_{y_2 \in \mathcal{Y}_2} P^{(2)}(y_2|x) P(y_1|y_2) \quad (17)$$

For channel coding we consider the Bhattacharyya parameter $Z(U_i|U_1^{i-1}, Y_1^N)$, where U_1^N is the vector of symbols at the input of the q -ary encoder G_N^{-1} , as indicated in Fig. 2. In the following claim we extend to the q -ary case the relation in [7] between channel degradation and Bhattacharyya parameters of binary channels.

Claim 2: Given two channels $P^{(1)}$ and $P^{(2)}$, with $P^{(1)} \preceq P^{(2)}$, then $Z(U_i|U_1^{i-1}, Y_1^N) \geq Z(U_i|U_1^{i-1}, Y_2^N)$.

Proof: See Appendix B.

The relation between channel degradation and the Bhattacharyya parameters of the coordinate channels [1] has been used to apply binary polar codes to relay channels [17], [18]. In [16] a rate adaptation technique is applied to Rayleigh channels with binary polar coding. With the help of Claim 2 it is straightforward to extend the commented binary applications to the q -ary case.

For polar channel coding on the AWGN we need to establish a map between \mathbb{F}_q and a set of signals. In this paper a signal is a complex number that represents a complex function of time. Therefore, for the constellation $C = \{t_j : j = 0, \dots, q-1; t_j \in \mathbb{C}\}$ we have the map $x_j \mapsto t_j$, $x_j \in \mathbb{F}_q$, $t_j \in \mathbb{C}$, $j = 0, \dots, q-1$.

Let the noise of the AWGN channel be the random variable Z , and $Z_r = \Re(Z)$, $Z_i = \Im(Z)$, with $\Re(\cdot)$ and $\Im(\cdot)$ the real and imaginary parts, respectively, of the argument, and let the

mean square of the noise be $E[|Z|^2] = 2\sigma^2$. The probability density function (pdf) of Z is [22]

$$f_Z(z) = f_{Z_r, Z_i}(z_r, z_i) = \frac{1}{2\pi\sigma^2} e^{-\frac{1}{2\sigma^2}(z_r^2 + z_i^2)}. \quad (18)$$

The signals at the output of the channel are, $Y_i = t(x_i) + Z_i$, $i = 1 \dots N$. As the symbols x_i , $x_i \in \mathbb{F}_q$, are equally likely, then the components of the LR vector for the first step of the iteration $\mathbf{L}_1^{(1)}(\cdot|y_i)$, $i = 1, \dots, N$ are

$$L_1^{(1)}(x|y_i) = \frac{f_Z(y_i - t(x))}{f_Z(y_i - t(0))}; x \in \mathbb{F}_q, x \neq 0 \quad (19)$$

The log LR (LLR) of (19) is,

$$\begin{aligned} l_1^{(1)}(x|y_i) &= \frac{\Re(t(x)) - \Re(t(0))}{\sigma^2} \left(\Re(y_i) - \frac{1}{2}(\Re(t(x)) + \Re(t(0))) \right) \\ &+ \frac{\Im(t(x)) - \Im(t(0))}{\sigma^2} \left(\Im(y_i) - \frac{1}{2}(\Im(t(x)) + \Im(t(0))) \right), \end{aligned} \quad (20)$$

where $x \in \mathbb{F}_q, x \neq 0$.

We consider two kind of constellations, rectangular shaped and circular shaped.

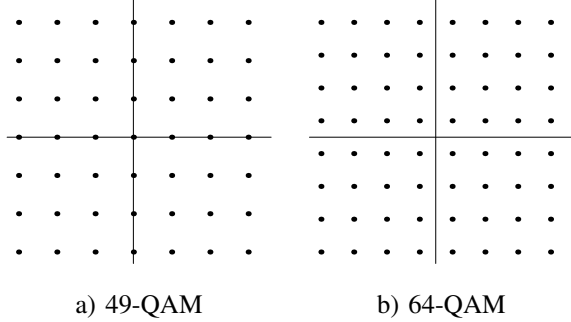


Fig. 3. Rectangular QAM constellations.

1) *Rectangular Shaped Constellations*: They are formed by spreading points equally spaced along the real and imaginary axis. We form a PAM constellation along each axis. If the number of points per axis is prime, the PAM constellation is, $C_{pPAM} = \{i - \lfloor p/2 \rfloor : i = 0, \dots, p-1\}$, where $\lfloor \cdot \rfloor$ is the floor function. If the size of the PAM constellation is 2^m , the constellation is, $C_{2^m PAM} = \{2i - (2^m + 1) : i = 1, \dots, 2^m\}$. It is possible to work with two different alternatives with rectangular shaped constellations. One alternative is to use two independent PAM constellations, each with its own polar code, one in the real axis and the other in the imaginary axis, in order to obtain the rectangular quadrature amplitude modulation (QAM) constellation. The main advantage of this alternative is its efficiency. The other alternative is to use all the points of the rectangular QAM constellation with one q-ary polar code. This alternative has better performance, but is less efficient than the previous one.

It is difficult to work with prime PAM constellations with size larger than 13 symbols, as it is shown in Section V. However, PAM constellation with size 2^m , $m \in \mathbb{N}$, do not have

the same problems and in Section V we work with two 32-PAM constellations to construct a 1024-QAM constellation. In Fig. 3, a) and b), we show the constellations 49-QAM and 64-QAM, respectively. Both of them have been used in the simulations.

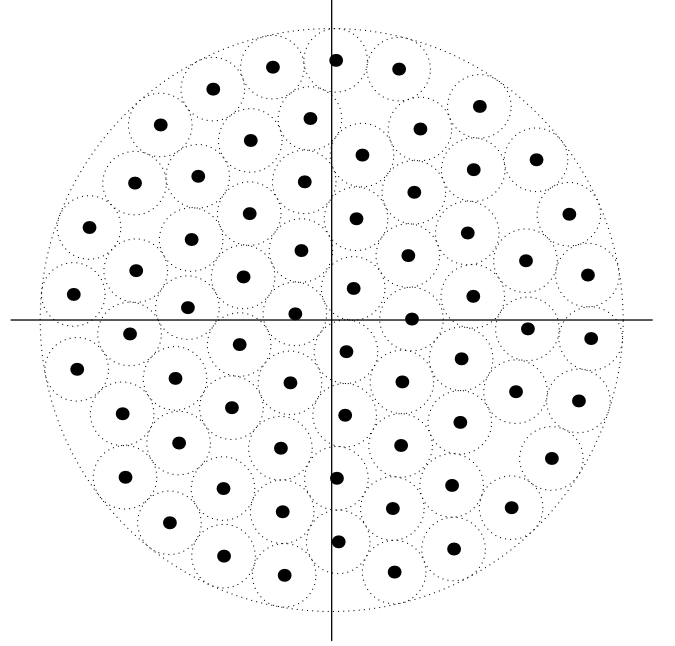


Fig. 4. Circular 67-QAM constellation.

2) *Circular Shaped Constellation*: The problem of maximizing the minimum pairwise distance between points of a constellation of a fixed energy, whose points are equally likely, is equivalent to the problem of optimal packing of equal circles in a circle, where optimal means non-overlapping circles of maximum radius [23]. In [24] a data base is maintained of the best packings of circles in a circle up to a size of 2600 circles. By choosing a packing with a number of circles equal to the order of a field we can define a constellation formed by the center of the circles. As the points of the constellation are spread in a circle we refer to it as circular QAM constellation. In Fig. 4 we show the circular 67-QAM constellation, that it is one of the constellations used in this paper. The interest of the 67-QAM constellation is more theoretical than practical, as a small rate loss is expected from mapping binary information to symbols. However, the results show that this constellation is close to the theoretical limit, compensating the commented rate loss. Additionally, efficiency is gained from working in a prime field, with addition module a prime, as compared to that in a non-prime field, where the addition is more complex.

C. Encoder and Decoder Implementation

The equations (1) and (8), that describe the q-ary polar encoder and decoder respectively, have a similar structure to the equations defining the binary polar encoder and decoder. Therefore the implementation of the q-ary polar encoder and decoder follows similar principles as in the binary case.

The decoder calculates the LR's (8) in order to carry out SC decoding. Eq. (8) is recursive and, in order to show the

dependency of the current LR on the previous ones, we express them as,

$$\begin{aligned}
\mathbf{L}_N^{(2i-1)} &= f(\mathbf{L}_{\frac{N}{2}}^{(i)}(u_{1,o}^{2i-2} - \alpha u_{1,e}^{2i-2}), \mathbf{L}_{\frac{N}{2}}^{(i)}(u_{1,e}^{2i-2})) \\
&= f(\mathbf{L}_{\frac{N}{2},oe}^{(i)}, \mathbf{L}_{\frac{N}{2},e}^{(i)}) \\
\mathbf{L}_N^{(2i)} &= g(\mathbf{L}_{\frac{N}{2}}^{(i)}(u_{1,o}^{2i-2} - \alpha u_{1,e}^{2i-2}), \mathbf{L}_{\frac{N}{2}}^{(i)}(u_{1,e}^{2i-2}, u_{2i-1})) \\
&= g(\mathbf{L}_{\frac{N}{2},oe}^{(i)}, \mathbf{L}_{\frac{N}{2},e}^{(i)}, u_{2i-1}),
\end{aligned} \tag{21}$$

where $\mathbf{L}_{\frac{N}{2},oe}^{(i)}$, $\mathbf{L}_{\frac{N}{2},e}^{(i)}$, $f()$ and $g()$ have been introduced to highlight the functional dependency of the LR in the actual recursion on the LR of the previous one.

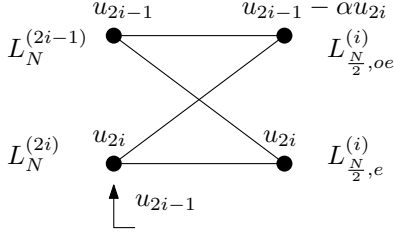


Fig. 5. Butterfly for decoding

The structure of (21) resembles that of the basic unit for calculating the fast Fourier transform (FFT): the butterfly structure. In Fig. 5 we represent the butterfly diagram of (21). A similar structure is also used for decoding binary polar codes [1]. There are two kind of calculations in (21) and in Fig. 5: from right to left, forward direction, the LR of the actual recursion are calculated with (21) as a function of the LR from the previous iteration. From left to right, backward direction, the extended transformation (1) is inverted. The even LR vector, $\mathbf{L}_{2N}^{(2i)}$, is a function of the symbol u_{2i-1} and this fact is indicated in Fig. 5 by the arrow in lower left side of the butterfly. The values calculated in the forward direction, two vectors of real numbers, and backward direction, two elements of \mathbb{F}_q , must be stored for their use by other butterflies.

We use an example to describe the algorithm implemented in the decoder. The codeword length is $N=8$ and the symbol alphabet is \mathbb{F}_3 , with elements $\{0,1,2\}$ and modular arithmetic modulus 3. In Fig. 6 we present the butterfly diagram used for decoding. On the right of the diagram are the signals y_i , $i = 1 \dots 8$, at the input of the decoder. They are applied to the decoder after a bit reversal permutation. The LR vectors of these signals are calculated with (12), or with (19), and they are the input to the rightmost butterflies. The detected symbols are on the left out of the diagram. Decoding starts with the first symbol, \hat{u}_1 , continues with the second when the first has been detected, and so on. The butterfly associated to the first symbol checks if it has LR vectors at its input. If it is the case the butterfly obtains the LR and performs the detection. Otherwise it transfers control to its neighboring butterflies on the right, first the upper one and, when it receives the LR vector, to the lower one. This process of transferring control is repeated from left to right until a butterfly is able to calculate the LR vector. In this case the butterfly gives the LR vector to its left neighbor and transfers control to it. When

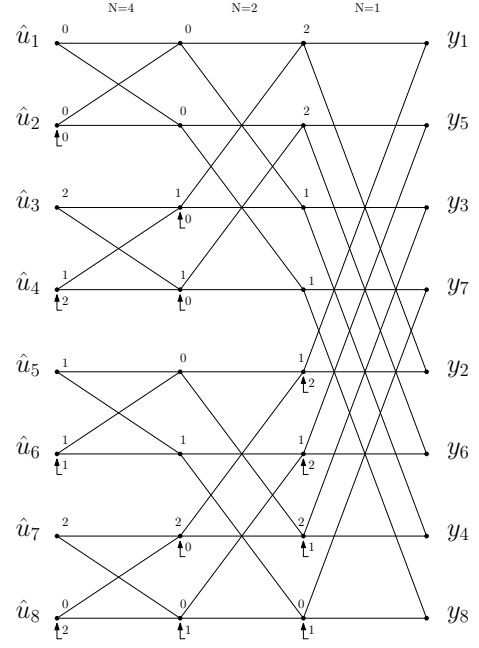


Fig. 6. Butterfly diagram

the upper leftmost butterfly has both LR vectors available it first obtains $L_8^{(1)}$ and detects the symbol \hat{u}_1 . With it the LR $L_8^{(2)}$ is calculated and \hat{u}_2 is detected. When both symbols of a butterfly are detected, or they are available, it does the inverse extended transformation (1). In Fig. 6 we have indicated the detected symbols on the left of the leftmost butterflies. On the right of the butterflies are the symbols obtained after inverting the extended transformation. These symbols are needed by the inner butterflies. For instance, the butterfly identified by its left upper corner as \hat{u}_1 and recursion $N = 2$, needs the symbol $\hat{u}_1 - \hat{u}_2$ (0 in this case) to obtain the even LR. Therefore, as the symbols \hat{u}_i , $i = 1 \dots N$, are detected, the extended transformation is inverted in the butterflies from left to right. As a result, the rightmost butterflies give the vector $\hat{\mathbf{u}}^8 G_8^{-1}$. Therefore, the butterfly diagram can be used for encoding too.

In this paper we have used a butterfly diagram for encoding. It avoids problems with the storage of the matrix G_N , or its inverse, and a better efficiency can be obtained as matrix multiplication is less efficient, in general.

D. Complexity

From (1) the complexity of the q -ary polar encoder is similar to the binary encoder: $N \log_2 N$. The butterfly diagram describes the complexity of the decoder. The number of butterflies is $\frac{N}{2} \times \log_2 N$. From (8), in each butterfly the odd LR needs q operations per LR coordinate. The even LR only needs one operation per coordinate. Therefore the complexity of decoding is $\mathcal{O}(\frac{q^2}{2} N \log_2 N)$.

For decoding each butterfly stores the calculated LR, with the exception of the leftmost butterflies: they do not store the obtained LR, they just detect. However, the LR at the input of the rightmost butterflies must be stored. The butterflies must also store the symbols obtained from the inverse transformation. If the memory for storing symbols

is considered negligible compared to that for storing real numbers, then the decoder needs $\mathcal{O}((p-1)N \log_2 N)$ cells for real numbers.

Many of the operations indicated in the butterfly diagram in Fig. 6 can be run in parallel. For instance, in all the rightmost butterflies the odd LRs $L_2^{(1)}$ can be calculated in parallel. In their neighboring butterflies, half of the odd LRs can be calculated in parallel, and so on. And the odd LRs are the most complex, as commented before. Likewise, the frozen symbols can help in the process of parallel computing. Obviously, the scheduling of the operations among butterflies must be adapted for parallelization.

V. NUMERICAL RESULTS

We have performed several numerical experiments in order to corroborate the theoretical results and to gain insight into the application of q-ary polar codes. Two kind of experiments have been performed: source coding simulations and channel coding for the AWGN channel.

A. Source Coding with Side Information

$P_X(X)$	x				
	0	1	2	3	4
	0.300	0.200	0.300	0.100	0.100

TABLE I
PROBABILITIES OF THE SOURCE SYMBOLS.

$P(X Y)$		y				
x	0	0.522	0.261	0.136	0.231	0.316
	1	0.261	0.348	0.092	0.154	0.105
	2	0.131	0.261	0.682	0.231	0.158
	3	0.043	0.087	0.045	0.307	0.105
	4	0.043	0.043	0.045	0.077	0.316

TABLE II
CONDITIONAL PROBABILITIES OF THE SOURCE SYMBOLS.

Several simulations have been done with several sources in order to know how close the results with finite codeword length are to the theoretical limits. In tables I and II we gather the probabilities associated to one of sources considered in the experiments. The conditional entropy of the source is $H(X|Y) = 1.90061$ bits. The source alphabet is \mathbb{F}_5 . Three codeword lengths have been chosen: $N = 4096$, $N = 16384$, and $N = 65536$. The codes have been constructed using Monte Carlo, as explained in Section III. The ordered Bhattacharyya parameters of the codes are presented in Fig. 7 as a function of the indices normalized by N . As expected, the longer the code is, the steeper the slope of the curve.

For the code construction the set \mathcal{A} was formed with indices whose Bhattacharyya sum is upper bounded,

$$\sum_{i \in \mathcal{A}} Z(U_i | Y_1^N, U_1^{i-1}) \leq 10^{-4}. \quad (22)$$

The source code rate R_s is obtained from $|\mathcal{A}|$. In Fig. 8 the source rate of the three mentioned codes is plotted

versus the codeword length. In the same figure $H(X|Y)$ is represented. The source rate gets closer to the entropy as the codeword length increases. Also, in Fig. 8 we present the symbol error rate, indicated by P_s in the figure, obtained by counting symbol errors between the input to the encoder and the output from the decoder. The upper bound 10^{-4} in (22), used for the code construction, has also been represented in Fig. 8. For the shortest code the simulation results are close to the sum upper bound. However, for long codes the criterion for constructing \mathcal{A} using the sum bound is conservative.

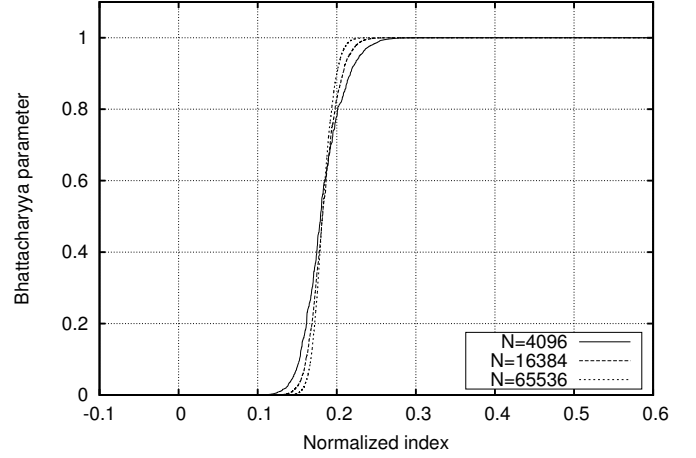


Fig. 7. Bhattacharyya parameters of the source polar codes.

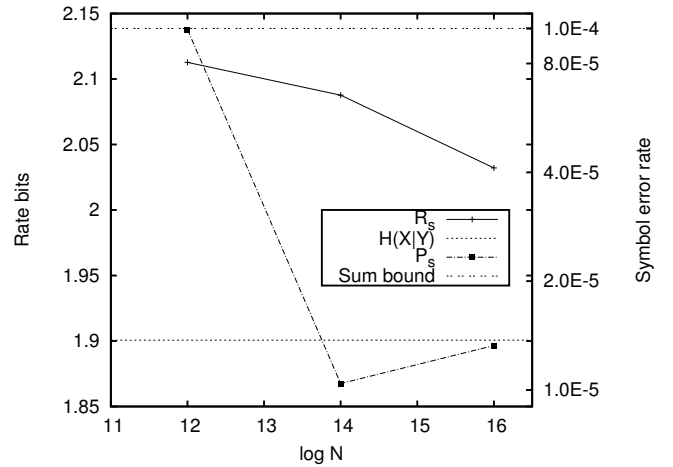


Fig. 8. Source compression rate of three codes constructed with the sum bound criterion in (22). The x-axis shows the \log_2 of the codeword length. In the same figure it is shown the symbol error rate obtained with the codes.

B. Channel Coding

A set of simulations have been done in order to know the performance of q-ary polar channel coding. Despite we have done simulations with both discrete and continuous channels, in this work we only present results for the continuous case: the AWGN channel. The signal noise ratio (SNR) has been defined as $SNR = E_s/E[|Z|^2]$, where $E_s = E[|t(x)|^2]$.

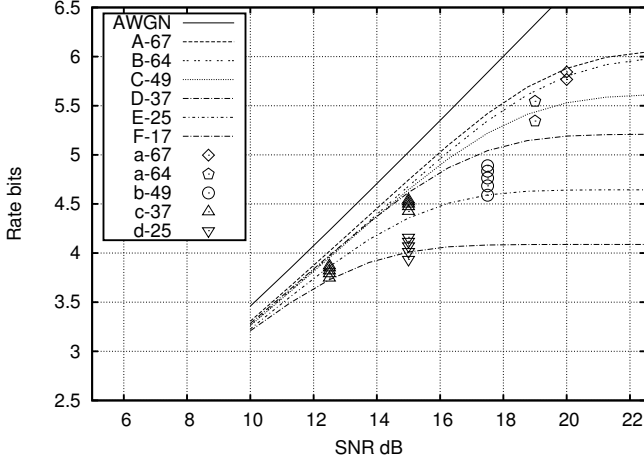


Fig. 9. Theoretical rates and rates obtained in the simulations versus SNR in dB. The curves labelled with appercase letters are theoretical. The number following the letter is the constellation's size. For curves a-64 and a-67 the codeword lengths are in $\{2048, 65536\}$. In the remaining simulations the codeword lengths are in $\{4096, 8192, 16384, 65536\}$.

The two constellations presented in Section IV have been considered. The code construction was done in a similar way as that for source coding, using the LR for obtaining the Bhattacharyya parameters, as indicated in Section III. Two channel coding alternatives have been used for rectangular QAM constellations. In one of them each PAM constellation has been considered independently for coding. In the other all the constellation points are considered for coding. For the independent PAM constellations the LLR (20) and the noise variance are adapted for the case of real signals. Different codeword lengths have been considered, from 2048 to 525288.

In the first set of experiments we obtained the information set \mathcal{A} for different constellations and codeword lengths. The criterion for including an index in \mathcal{A} was that its Bhattacharyya parameter were less than a prefixed value, instead of the sum Bhattacharyya parameters used for source coding. The chosen value was 10^{-4} . In Fig. 9 we show the rates $R = R_c \log_2 q$, in bits, obtained with simulation for different codeword lengths and SNRs. For comparison, in the same figure we show the theoretical mutual information curves associated to the constellations with equally likely constellation points. The curves have been calculated numerically. The constellations are identified by their cardinality; if it is prime the constellation's shape is circular, otherwise it is rectangular. In the case of rectangular QAM constellations the coding has been performed on each PAM constellation independently. The theoretical results are indicated with uppercase letters. Two sets of codeword lengths have been considered for the simulations $\{2048, 65536\}$ for 64 and 67 signal points per constellation and $\{4096, 8192, 16384, 65536\}$ for the remaining constellations. Increasing the codeword length for a fixed SNR and constellation, renders rates that are closer to the theoretical results. Because of this we have not labeled the codeword lengths: they are easily identified by their position in the pile corresponding to the SNR and shape.

Three main conclusions can be extracted from the results

in Fig. 9, a) the circular QAM shaped constellation gives results that are close to the theoretical curves, even with short codeword lengths, b) the rectangular QAM constellation with size 2^m , $m \in \mathbb{N}$, is close to the theoretical curves for short codes, though the separation from the theoretical curves is noticeable, c) the rectangular QAM shaped constellation with size p^2 , p prime, is further from the theoretical results and the gap increases with the constellation size.

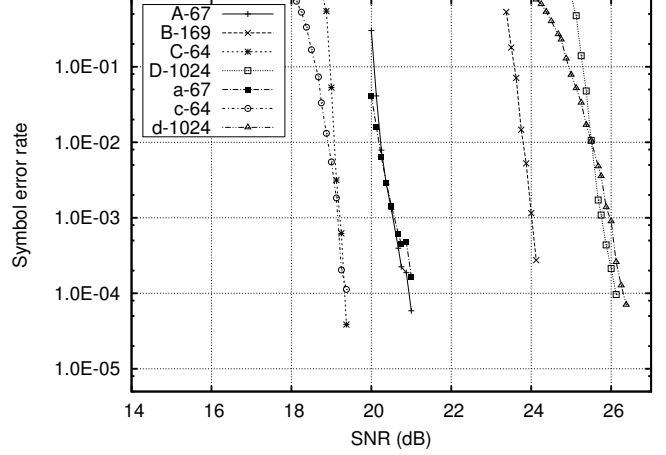


Fig. 10. Results of symbol error rate versus relative SNR obtained simulating with different constellations and codeword lengths. The constellations are indicated by their size in the labels of the curves. With uppercase we indicate long codes, $N = 524288$ for the 169-QAM constellation and $N = 65536$ for the rest of long codes. The short codes, $N=2048$, are labeled with lower case letters. The circular constellation and codes A-67 and a-67 where designed for a target $SNR = 20.0$ dB, the rectangular B-169 for $SNR = 23$ dB, the rectangular C-64 and c-64 for $SNR = 19$ dB, and the rectangular D-1024 and d-1024 for $SNR = 25$ dB.

In Fig. 10 we represent symbol error rate versus SNR measured by counting errors between the input to the encoder and the output from the decoder. The rectangular QAM constellation uses independent coding in each PAM constellation. The constellation's size varies from 64 to 1024 points and the codeword length from 2048 to 524288. The codes were constructed with the criteria commented above and the channel code rates were $R_c = 0.9631$ and $R_c = 0.9507$ for codes A-67 and a-67, respectively; $R_c = 0.9242$ and $R_c = 0.9801$ for codes C-64 and c-64 respectively; $R_c = 0.7549$ and $R_c = 0.707$ for codes D-1024 and d-1024 respectively, and finally, $R_c = 0.9030$ for the code B-169.

A first conclusion from the results shown in Fig. 10 is that rectangular constellations with sizes p^2 , p prime, need very long codes in order to have error rate figures similar to that obtained with the other constellations considered in this paper. They use prime sized PAM constellations and therefore the polar codes use the basic polar transformation [1]. The simulations show that the curves of ordered Bhattacharyya parameters versus index number, corresponding to PAM constellations with prime size, do not have a fast fall when the constellation size is larger than 7. This problem does not appear with PAM constellations with size 2^m , $m \in \mathbb{N}$, where the extended polar transformation (1) is used. In fact, it is possible to construct very large QAM constellations,

as shown in curves D-1024 and d-1024, working with PAM constellations of size 2^m , $m \in \mathbb{N}$, coding with the extended polar transformation (1).

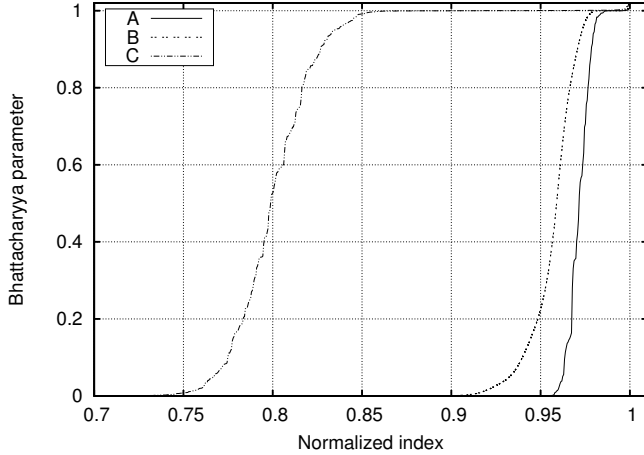


Fig. 11. Ordered Bhattacharyya parameters versus the indices normalized by the codeword length of three of the codes considered in Fig. 10. The curve A is for the polar code with a circular constellation with 67 points and codeword length $N = 2048$. The curve B is for the polar code used with a rectangular constellation of 169 points and codeword length $N = 524288$. In curve C the codeword length is $N = 2048$ with a rectangular 1024-QAM constellation.

In Fig. 11 we compare the Bhattacharyya parameters of the longest and the shortest polar codes of Fig. 10. The curve A in Fig. 11, corresponding to the short polar code with circular constellation, has a faster fall than that of the longer codes. The shape of the curves B and C is similar in the region of low Bhattacharyya parameters. This is interesting and shows that with rectangular constellations and relatively short codes, and coding with the extended transformation (1), it is possible to obtain symbol error figures similar to the ones obtained with rectangular constellations and the basic transformation, but much longer codeword lengths.

C. Comparison with Other Techniques

We have compared the direct q-ary polar approach presented in this work with the two main coding techniques for Gaussian channels with multilevel signals: Multilevel coding (MLC) [14] and bit-interleaved coded modulation (BICM) [25]. We have used [6] as a reference for comparison because it gathers results obtained with multilevel polar coding (MLPC) and BICM using binary polar codes. Therefore it is possible to compare the results obtained using binary polar codes for MLPC or BICM with the more direct approach here presented. Furthermore, there is a comparison in [6] between MLPC and the BICM scheme of the DVB-T2 standard [26]. This comparison is an indication of how adequate are polar codes for real applications.

Three signal sets are considered with sizes $M=4$, $M=16$, and $M=256$. The codeword length of the concatenated LDPC and BCH code of the DVB-T2 standard is $N=64800$. In [6] the overall length of the MLPC used in the comparisons was $N=65536$. The codeword lengths of the component binary

polar codes were: $N=32768$ for $M=4$, $N=16384$ for $M=16$ and $N=8192$ for $M=256$. We have chosen the same codeword lengths for the q-ary polar codes in order to make a fair comparison. The maximum word error rate (WER) was $WER_{\max}=10^{-7}$. Labeling is important for codes with MLC in order to obtain good codes [14], [6]. Set-partitioning labeling was the choice for the MLPC used for comparison with the BICM scheme of DVB-T2 [6]. However, no optimization of labeling is needed for q-ary polar codes.

In Fig. 12 we present the results of the comparison. In the x-axis the SNR and in the y-axis the rate in bits per symbol. The data of the MLPC and the BICM of DVB-T2 are from [6, Fig. 12]. The rates of the MLPC are obtained from density evolution (DE). However, it is a difficult task to implement DE with q-ary polar codes. The low value of WER required by DVB-T2 makes difficult to obtain information of rate from counting errors in the simulations. The bound (15) has been used to obtain the information set. If the information set \mathcal{A} is formed with the set of indices such that $(q-1) \sum_{i \in \mathcal{A}} Z(U_i | Y_1^N, U_1^{i-1}) \leq 10^{-7}$ then $WER \leq 10^{-7}$. The results from [6] come from DE, they are approximate; however, the q-ary polar code results are obtained from a worst case scenario. In Fig. 12 we have included the rectangular shaped constellation, labelled as RQAM in the curves, and the circular shaped constellation, label CQAM. One q-ary polar code with all the constellation points has been used with the rectangular QAM constellations. The curve with label DVB-T2 indicates the reference values for DVB-T2 and the curve with label MLPC corresponds to the MLPC scheme with data from [6]. The marks on the DVB-T2 curves indicate the size of the signal sets. We have not put marks on the other curves because it is easy to identify them as they are close to the DVB-T2 ones.

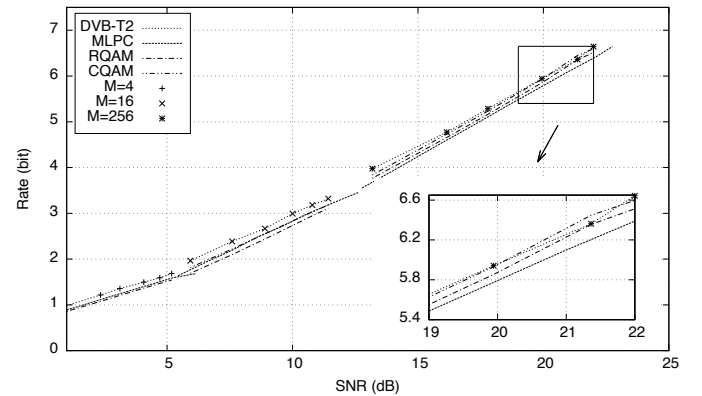


Fig. 12. Comparison of multilevel polar codes, BICM codes of DVB-T2 and q-ary polar codes.

The results in Fig. 12 show that for small signal sets, $M=4$ and $M=16$, MLPC is slightly better than q-ary polar coding with rectangular shaped constellation, curves with label RQAM. For $M=16$ the curves of MLPC and q-ary polar coding with circular constellations, curve CQAM, coincide. Both coding systems, MLPC and q-ary polar coding, have curves below that of the DVB-T2. However, for large signal sets, $M=256$, q-ary polar codes with both rectangular and

circular shaped constellations have curves above the MLPC one. Furthermore, with q-ary polar coding and circular shaped constellation it is possible to approach the DVB-T2 objectives working with high SNR.

We have not made comparisons with BICM schemes with polar codes as, according to [6], they operate further from capacity than the MLPC schemes. Using the results in [14] we have compared a MLC with a turbo code as component code against q-ary polar codes. The TC codeword length was $N=20000$. Other conditions were: $\text{SNR}=16.865$ dB, bit error rate (BER) $=10^{-5}$, signal set size $M=64$. The rate was [14] $R=5$ bits. With a q-ary polar code and rectangular shaped constellation, codeword length $N=16384$, with same SNR, M , and BER, the rate was $R=5.00$ bits. With a circular shaped constellation the rate was $R=5.05$ bits. With a q-ary polar code, rectangular shaped constellation, and $N=32768$, the rate was $R=5.04$ bits.

The complexity of q-ary polar decoding is quadratic in the alphabet size. The decoding complexity of MLPC is linear in the alphabet size. However, q-ary polar decoding is more adequate for a parallel implementation and can be implemented with the current technology. The comparison of the complexity of q-ary polar coding with MLC with an arbitrary component code is more involved as it depends in the particular component code.

A summary of the comparisons done with other codes is the following: for 1D signal sets MLC and MLPC are slightly better, i.e., closer to capacity, than q-ary polar codes. For 2D signal sets with small size, same results as for 1D case. However, for 2D signal sets with size 64 and over, q-ary polar codes are better than MLC or MLPC with both circular and rectangular constellations.

VI. CONCLUSIONS

In this paper we have presented a direct approach to the problem of q-ary polar encoding and decoding for sources and channels. The encoding process is very efficient. We express the LR for decoding in a recursive equation by means of a LR vector, allowing the implementation of SC decoding in a way that is similar to the binary case. The decoding algorithm is quadratic in the size of the field used for coding, however, the LR vector softens the complexity by a factor of $1/2$ and many of the more complex operations can be parallelized. The Bhattacharyya parameters can be put as a function of the decoding LR allowing the construction of the code using Monte Carlo, as with the binary case. We have done numerical experiments of direct q-ary polar coding for sources and channels with alphabets and constellations of sizes from 5 to 1024 and codeword lengths between 2048 and 524288. For the Gaussian channel we have used rectangular QAM constellations and optimal circular constellations. We have also compared the direct q-ary polar coding with other multilevel schemes for channel coding. The results show that direct q-ary polar coding is closer to the theoretical limit than other alternatives when the alphabet size is large. Therefore it is possible to apply direct q-ary polar coding to real scenarios with relatively short codes.

APPENDIX A

PROOF OF CLAIM 1

The extended polar transformation (1) is invertible and so is the polar transform from X_1^N to U_1^N shown in Fig. 1. Therefore we have,

$$\begin{aligned} P_{N U_1^N | Y_1^N}(u_1^N | y_1^N) \\ = P_{\frac{N}{2} S_1^{\frac{N}{2}} | Y_1^{\frac{N}{2}}}(s_1^{\frac{N}{2}} | y_1^{\frac{N}{2}}) P_{\frac{N}{2} S_{\frac{N}{2}+1}^{\frac{N}{2}} | Y_{\frac{N}{2}+1}^{\frac{N}{2}}}(s_{\frac{N}{2}+1}^{\frac{N}{2}} | y_{\frac{N}{2}+1}^{\frac{N}{2}}). \end{aligned} \quad (23)$$

In order to alleviate the notation, hereafter we omit the subscripts from the probabilities when they are obvious from the context.

From (23) and Fig. 1 we have,

$$P_N(u_1^N | y_1^N) = P_N(u_{1,o}^N - \alpha u_{1,e}^N | y_1^{\frac{N}{2}}) P_N(u_{1,e}^N | y_{\frac{N}{2}+1}^{\frac{N}{2}}), \quad (24)$$

where the subscripts 1, o , or 1, e , indicate the odd, or even, indices, respectively, of the vector and the minus sign refers to the sum of the additive inverse element in \mathbb{F}_q .

We use (24) to obtain,

$$\begin{aligned} P_N(u_1^{2i-1} | y_1^N) \\ = \sum_{u_{2i,o}^N} \sum_{u_{2i,e}^N} P_{\frac{N}{2}}(u_{1,o}^N - \alpha u_{1,e}^N | y_1^{\frac{N}{2}}) P_{\frac{N}{2}}(u_{1,e}^N | y_{\frac{N}{2}+1}^{\frac{N}{2}}) \\ = \sum_{u_{2i}^N} \sum_{u_{2i+1,e}^N} P_{\frac{N}{2}}(u_{1,e}^N | y_{\frac{N}{2}+1}^{\frac{N}{2}}) \underbrace{\sum_{u_{2i+1,o}^N} P_{\frac{N}{2}}(u_{1,o}^N - \alpha u_{1,e}^N | y_1^{\frac{N}{2}})}_{P_{\frac{N}{2}}(u_{1,o}^{2i} - \alpha u_{1,e}^{2i} | y_1^{\frac{N}{2}})} \\ = \sum_{u_{2i}^N} P_{\frac{N}{2}}(u_{1,o}^{2i} - \alpha u_{1,e}^{2i} | y_1^{\frac{N}{2}}) \underbrace{\sum_{u_{2i+1,e}^N} P_{\frac{N}{2}}(u_{1,e}^N | y_{\frac{N}{2}+1}^{\frac{N}{2}})}_{P_{\frac{N}{2}}(u_{1,e}^{2i} | y_{\frac{N}{2}+1}^{\frac{N}{2}})} \\ = \sum_{u_{2i}^N} P_{\frac{N}{2}}(u_{1,o}^{2i} - \alpha u_{1,e}^{2i} | y_1^{\frac{N}{2}}) P_{\frac{N}{2}}(u_{1,e}^{2i} | y_{\frac{N}{2}+1}^{\frac{N}{2}}). \end{aligned} \quad (25)$$

Similarly,

$$\begin{aligned} P_N(u_1^{2i} | y_1^N) \\ = \sum_{u_{2i+1,e}^N} P_{\frac{N}{2}}(u_{1,e}^N | y_{\frac{N}{2}+1}^{\frac{N}{2}}) \underbrace{\sum_{u_{2i+1,o}^N} P_{\frac{N}{2}}(u_{1,o}^N - \alpha u_{1,e}^N | y_1^{\frac{N}{2}})}_{P_{\frac{N}{2}}(u_{1,o}^{2i} - \alpha u_{1,e}^{2i} | y_1^{\frac{N}{2}})} \\ = P_{\frac{N}{2}}(u_{1,o}^{2i} - \alpha u_{1,e}^{2i} | y_1^{\frac{N}{2}}) P_{\frac{N}{2}}(u_{1,e}^{2i} | y_{\frac{N}{2}+1}^{\frac{N}{2}}). \end{aligned} \quad (26)$$

The conditional probability of the symbol u_{2i-1} is obtained from (24)

$$\begin{aligned} P_N^{(2i-1)}(u_{2i-1} | y_1^N, u_1^{2i-2}) \\ = \frac{1}{P_N(u_1^{2i-2} | y_1^N)} P_N(u_1^{2i-1} | y_1^N) \\ = \frac{\sum_{u_{2i}} P_{\frac{N}{2}}(u_{1,o}^{2i} - \alpha u_{1,e}^{2i} | y_1^{\frac{N}{2}}) P_{\frac{N}{2}}(u_{1,e}^{2i} | y_{\frac{N}{2}+1}^{\frac{N}{2}})}{P_N(u_1^{2i-2} | y_1^N)}, \end{aligned} \quad (27)$$

where we have used the superscript $(2i-1)$ and the subscript N to highlight the similarity of this probability with the coordinate channel defined in [1]. Using the fact,

$$\begin{aligned} & P_{\frac{N}{2}}(u_{1,o}^{2i} - \alpha u_{1,e}^{2i} | y_1^{\frac{N}{2}}) \\ &= P_{\frac{N}{2}}^{(2i-1)}(u_{2i-1} - \alpha u_{2i} | y_1^{\frac{N}{2}}, u_{1,0}^{2i-2} - \alpha u_{1,e}^{2i-2}) \\ &\quad \times P_{\frac{N}{2}}(u_{1,o}^{2i-2} - \alpha u_{1,e}^{2i-2} | y_1^{\frac{N}{2}}) \\ & P_{\frac{N}{2}}(u_{1,e}^{2i} | y_{\frac{N}{2}+1}^N) \\ &= P_{\frac{N}{2}}^{(2i)}(u_{2i} | y_{\frac{N}{2}+1}^N, u_{1,e}^{2i-2}) P_{\frac{N}{2}}(u_{1,e}^{2i-2} | y_{\frac{N}{2}+1}^N), \end{aligned}$$

we can obtain from (27) the following expression for $P_N^{(2i-1)}(u_{2i-1} | y_1^N, u_1^{2i-2})$,

$$\begin{aligned} & P_N^{(2i-1)}(u_{2i-1} | y_1^N, u_1^{2i-2}) \\ &= \frac{P_{\frac{N}{2}}(u_{1,o}^{2i-2} - \alpha u_{1,e}^{2i-2} | y_1^{\frac{N}{2}}) P_{\frac{N}{2}}(u_{1,e}^{2i-2} | y_{\frac{N}{2}+1}^N)}{P_N(u_1^{2i-2} | y_1^N)} \\ &\quad \times \sum_{u_{2i}} P_{\frac{N}{2}}^{(i)}(u_{2i-1} - \alpha u_{2i} | y_1^{\frac{N}{2}}, u_{1,0}^{2i-2} - \alpha u_{1,e}^{2i-2}) \\ &\quad \times P_{\frac{N}{2}}^{(i)}(u_{2i} | y_{\frac{N}{2}+1}^N, u_{1,e}^{2i-2}). \end{aligned} \quad (29)$$

Using similar arguments we have,

$$\begin{aligned} & P_N^{(2i)}(u_{2i} | y_1^N, u_1^{2i-1}) \\ &= \frac{P_{\frac{N}{2}}(u_{1,o}^{2i-1} - \alpha u_{1,e}^{2i-1} | y_1^{\frac{N}{2}}) P_{\frac{N}{2}}(u_{1,e}^{2i-1} | y_{\frac{N}{2}+1}^N)}{P_N(u_1^{2i-1} | y_1^N)} \\ &\quad \times P_{\frac{N}{2}}^{(i)}(u_{2i-1} - \alpha u_{2i} | y_1^{\frac{N}{2}}, u_{1,0}^{2i-2} - \alpha u_{1,e}^{2i-2}) \\ &\quad \times P_{\frac{N}{2}}^{(i)}(u_{2i} | y_{\frac{N}{2}+1}^N, u_{1,e}^{2i-2}). \end{aligned} \quad (30)$$

From (29) we can express $L_N^{(2i-1)}(u | y_1^N, u_1^{2i-2})$ as,

$$\begin{aligned} & L_N^{(2i-1)}(u | y_1^N, u_1^{2i-2}) \\ &= \frac{P_N^{(2i-1)}(u | y_1^N, u_1^{2i-2})}{P_N^{(2i-1)}(0 | y_1^N, u_1^{2i-2})} = \\ & \frac{\sum_{u_{2i}} P_{\frac{N}{2}}^{(i)}(u - \alpha u_{2i} | y_1^{\frac{N}{2}}, u_{1,0}^{2i-2} - \alpha u_{1,e}^{2i-2}) P_{\frac{N}{2}}^{(i)}(u_{2i} | y_{\frac{N}{2}+1}^N, u_{1,e}^{2i-2})}{\sum_{u_{2i}} P_{\frac{N}{2}}^{(i)}(-\alpha u_{2i} | y_1^{\frac{N}{2}}, u_{1,0}^{2i-2} - \alpha u_{1,e}^{2i-2}) P_{\frac{N}{2}}^{(i)}(u_{2i} | y_{\frac{N}{2}+1}^N, u_{1,e}^{2i-2})}, \\ & \quad u \in \mathbb{F}_q, u \neq 0. \end{aligned} \quad (31)$$

After dividing the numerator and denominator of the last term of (31) by

$$P_{\frac{N}{2}}^{(i)}(0 | y_1^{\frac{N}{2}}, u_{1,0}^{2i-2} - \alpha u_{1,e}^{2i-2}) P_{\frac{N}{2}}^{(i)}(0 | y_{\frac{N}{2}+1}^N, u_{1,e}^{2i-2})$$

we obtain the first result of (8). The second result is obtained in a similar way. First from (30) we have,

$$\begin{aligned} & L_N^{(2i)}(u | y_1^N, u_1^{2i-2}, u_{2i-1}) \\ &= \frac{P_N^{(2i)}(u | y_1^N, u_1^{2i-1})}{P_N^{(2i)}(0 | y_1^N, u_1^{2i-1})} \\ &= \frac{P_{\frac{N}{2}}^{(i)}(u_{2i-1} - \alpha u | y_1^{\frac{N}{2}}, u_{1,0}^{2i-2} - \alpha u_{1,e}^{2i-2}) P_{\frac{N}{2}}^{(i)}(u | y_{\frac{N}{2}+1}^N, u_{1,e}^{2i-2})}{P_{\frac{N}{2}}^{(i)}(u_{2i-1} | y_1^{\frac{N}{2}}, u_{1,0}^{2i-2} - \alpha u_{1,e}^{2i-2}) P_{\frac{N}{2}}^{(i)}(0 | y_{\frac{N}{2}+1}^N, u_{1,e}^{2i-2})} \\ & \quad u \in \mathbb{F}_q, u \neq 0. \end{aligned} \quad (32)$$

We obtain the second result in (8) by dividing the numerator and denominator of (32) by

$$P_{\frac{N}{2}}^{(i)}(0 | y_1^{\frac{N}{2}}, u_{1,0}^{2i-2} - \alpha u_{1,e}^{2i-2})$$

APPENDIX B

PROOF OF CLAIM 2

The Bhattacharyya parameter associated with $P^{(1)}$ is

$$\begin{aligned} Z(U_i | Y_{11}^N, U_1^{i-1}) &= \frac{1}{q-1} \sum_{\substack{u_i, u'_i \\ u_i \neq u'_i}} \sum_{u_1^{i-1}} \sum_{y_1^N} \\ & \sqrt{P^{(1)}(u_i, u_1^{i-1}, y_1^N) P^{(1)}(u'_i, u_1^{i-1}, y_1^N)} \\ &= \frac{1}{q-1} \sum_{\substack{u_i, u'_i \\ u_i \neq u'_i}} \sum_{u_1^{i-1}} \sum_{y_1^N} \\ & \sqrt{\sum_{u_{i+1}^N} P^{(1)}(y_1^N | u_i, u_1^{i-1}, u_{i+1}^N) P_N(u_i, u_1^{i-1}, u_{i+1}^N)} \\ & \quad \times \sqrt{\sum_{u_{i+1}^N} P^{(1)}(y_1^N | u'_i, u_1^{i-1}, u_{i+1}^N) P_N(u'_i, u_1^{i-1}, u_{i+1}^N)}. \end{aligned} \quad (33)$$

Matrix G_N is invertible. We call S_1, S_i, S_{i+1} the linear subspaces spanned by the rows 1 to $i-1$, i , and $i+1$ to N , respectively, of G_N^{-1} . There is a one-to-one correspondence between $(u_1^{i-1}, u_i, u_{i+1}^N)$ and $\mathbf{x}_{u_i} + \mathbf{x}_1 + \mathbf{x}_2$, where $\mathbf{x}_1 = (u_1^{i-1}, 0, 0_{i+1}^N) G_N^{-1}$, $\mathbf{x}_{u_i} = (0, u_i, 0_{i+1}^N) G_N^{-1}$, and $\mathbf{x}_2 = (0_1^{i-1}, 0, u_{i+1}^N) G_N^{-1}$. Therefore Eq. (33) is

$$\begin{aligned} Z(U_i | Y_{11}^N, U_1^{i-1}) &= \frac{1}{q-1} \sum_{\substack{\mathbf{x}_{u_i}, \mathbf{x}_{u'_i} \in S_i \\ \mathbf{x}_{u_i} \neq \mathbf{x}_{u'_i}}} \sum_{\mathbf{x}_1 \in S_1} \sum_{y_1^N} \\ & \sqrt{\sum_{\mathbf{x}_2 \in S_{i+1}} P^{(1)}(y_1^N | \mathbf{x}_{u_i} + \mathbf{x}_1 + \mathbf{x}_2) P_N(\mathbf{x}_{u_i} + \mathbf{x}_1 + \mathbf{x}_2)} \\ & \quad \times \sqrt{\sum_{\mathbf{x}_2 \in S_{i+1}} P^{(1)}(y_1^N | \mathbf{x}_{u'_i} + \mathbf{x}_1 + \mathbf{x}_2) P_N(\mathbf{x}_{u'_i} + \mathbf{x}_1 + \mathbf{x}_2)} \end{aligned}$$

$$\begin{aligned}
&= \frac{1}{q-1} \sum_{\substack{\mathbf{x}_{u_i}, \mathbf{x}_{u'_i} \in S_i \\ \mathbf{x}_{u_i} \neq \mathbf{x}_{u'_i}}} \sum_{\mathbf{x}_1 \in S_1} \sum_{y_{11}^N} \\
&\left(\sum_{\mathbf{x}_2 \in S_{i+1}} P_N(\mathbf{x}_{u_i} + \mathbf{x}_1 + \mathbf{x}_2) \right. \\
&\quad \left. \prod_{j=1}^N P^{(1)}(y_{1j} | \mathbf{x}_{u_i j} + \mathbf{x}_{1j} + \mathbf{x}_{2j}) \right)^{1/2} \\
&\times \left(\sum_{\mathbf{x}_2 \in S_{i+1}} P_N(\mathbf{x}_{u'_i} + \mathbf{x}_1 + \mathbf{x}_2) \right. \\
&\quad \left. \prod_{j=1}^N P^{(1)}(y_{1j} | \mathbf{x}_{u'_i j} + \mathbf{x}_{1j} + \mathbf{x}_{2j}) \right)^{1/2}, \quad (34)
\end{aligned}$$

where we have used the fact that the channel is memoryless and \mathbf{x}_{aj} is the j -th component of \mathbf{x}_a .

By hypothesis $P^{(1)} \preceq P^{(2)}$. From (17) $P^{(1)}$ is a function of $P^{(2)}$ and a channel P . Using this fact in (34) we have,

$$\begin{aligned}
Z(U_i | Y_{11}^N, U_1^{i-1}) &= \frac{1}{q-1} \sum_{\substack{\mathbf{x}_{u_i}, \mathbf{x}_{u'_i} \in S_i \\ \mathbf{x}_{u_i} \neq \mathbf{x}_{u'_i}}} \sum_{\mathbf{x}_1 \in S_1} \sum_{y_{11}^N} \\
&\left(\sum_{\mathbf{x}_2 \in S_{i+1}} P_N(\mathbf{x}_{u_i} + \mathbf{x}_1 + \mathbf{x}_2) \right. \\
&\quad \left. \prod_{j=1}^N \sum_{y_{2j}} P^{(2)}(y_{2j} | \mathbf{x}_{u_i j} + \mathbf{x}_{1j} + \mathbf{x}_{2j}) P(y_{1j} | y_{2j}) \right)^{1/2} \\
&\times \left(\sum_{\mathbf{x}_2 \in S_{i+1}} P_N(\mathbf{x}_{u'_i} + \mathbf{x}_1 + \mathbf{x}_2) \right. \\
&\quad \left. \prod_{j=1}^N \sum_{y_{2j}} P^{(2)}(y_{2j} | \mathbf{x}_{u'_i j} + \mathbf{x}_{1j} + \mathbf{x}_{2j}) P(y_{1j} | y_{2j}) \right)^{1/2}. \quad (35)
\end{aligned}$$

By expanding the products in (35) we have,

$$\begin{aligned}
Z(U_i | Y_{11}^N, U_1^{i-1}) &= \frac{1}{q-1} \sum_{\substack{\mathbf{x}_{u_i}, \mathbf{x}_{u'_i} \in S_i \\ \mathbf{x}_{u_i} \neq \mathbf{x}_{u'_i}}} \sum_{\mathbf{x}_1 \in S_1} \sum_{y_{11}^N} \\
&\left(\sum_{\mathbf{x}_2 \in S_{i+1}} \sum_{y_{21}} \cdots \sum_{y_{2N}} P_N(\mathbf{x}_{u_i} + \mathbf{x}_1 + \mathbf{x}_2) \right. \\
&\quad \left. \prod_{j=1}^N P^{(2)}(y_{2j} | \mathbf{x}_{u_i j} + \mathbf{x}_{1j} + \mathbf{x}_{2j}) P(y_{1j} | y_{2j}) \right)^{1/2} \\
&\times \left(\sum_{\mathbf{x}_2 \in S_{i+1}} \sum_{y_{21}} \cdots \sum_{y_{2N}} P_N(\mathbf{x}_{u'_i} + \mathbf{x}_1 + \mathbf{x}_2) \right. \\
&\quad \left. \prod_{j=1}^N P^{(2)}(y_{2j} | \mathbf{x}_{u'_i j} + \mathbf{x}_{1j} + \mathbf{x}_{2j}) P(y_{1j} | y_{2j}) \right)^{1/2}
\end{aligned}$$

$$\begin{aligned}
&\stackrel{(a)}{\geq} \frac{1}{q-1} \times \sum_{\substack{\mathbf{x}_{u_i}, \mathbf{x}_{u'_i} \in S_i \\ \mathbf{x}_{u_i} \neq \mathbf{x}_{u'_i}}} \sum_{\mathbf{x}_1 \in S_1} \sum_{y_{11}^N} \sum_{y_{21}} \cdots \sum_{y_{2N}} \\
&\left(\sum_{\mathbf{x}_2 \in S_{i+1}} P_N(\mathbf{x}_{u_i} + \mathbf{x}_1 + \mathbf{x}_2) \right. \\
&\quad \left. \prod_{j=1}^N P^{(2)}(y_{2j} | \mathbf{x}_{u_i j} + \mathbf{x}_{1j} + \mathbf{x}_{2j}) P(y_{1j} | y_{2j}) \right)^{1/2} \\
&\times \left(\sum_{\mathbf{x}_2 \in S_{i+1}} P_N(\mathbf{x}_{u'_i} + \mathbf{x}_1 + \mathbf{x}_2) \right. \\
&\quad \left. \prod_{j=1}^N P^{(2)}(y_{2j} | \mathbf{x}_{u'_i j} + \mathbf{x}_{1j} + \mathbf{x}_{2j}) P(y_{1j} | y_{2j}) \right)^{1/2}, \quad (36)
\end{aligned}$$

where (a) follows from the Schwarz's inequality. The sum of the factor $P(y_{1j} | y_{2j})$ over all possible values of y_{1j} is one, and therefore (36) is

$$\begin{aligned}
&Z(U_i | Y_{11}^N, U_1^{i-1}) \\
&\geq \frac{1}{q-1} \sum_{\substack{\mathbf{x}_{u_i}, \mathbf{x}_{u'_i} \in S_i \\ \mathbf{x}_{u_i} \neq \mathbf{x}_{u'_i}}} \sum_{\mathbf{x}_1 \in S_1} \sum_{y_{21}} \cdots \sum_{y_{2N}} \\
&\left(\sum_{\mathbf{x}_2 \in S_{i+1}} P_N(\mathbf{x}_{u_i} + \mathbf{x}_1 + \mathbf{x}_2) \right. \\
&\quad \left. \prod_{j=1}^N P^{(2)}(y_{2j} | \mathbf{x}_{u_i j} + \mathbf{x}_{1j} + \mathbf{x}_{2j}) \right)^{1/2} \\
&\times \left(\sum_{\mathbf{x}_2 \in S_{i+1}} P_N(\mathbf{x}_{u'_i} + \mathbf{x}_1 + \mathbf{x}_2) \right. \\
&\quad \left. \prod_{j=1}^N P^{(2)}(y_{2j} | \mathbf{x}_{u'_i j} + \mathbf{x}_{1j} + \mathbf{x}_{2j}) \right)^{1/2} \\
&= \frac{1}{q-1} \sum_{\substack{\mathbf{x}_{u_i}, \mathbf{x}_{u'_i} \in S_i \\ \mathbf{x}_{u_i} \neq \mathbf{x}_{u'_i}}} \sum_{\mathbf{x}_1 \in S_1} \sum_{y_{11}^N} \\
&\left(P^{(2)}_N(y_{21}^N, \mathbf{x}_{u_i} + \mathbf{x}_1) \right)^{1/2} \\
&\times \left(\prod_{j=1}^N P^{(2)}_N(y_{2j}^N, \mathbf{x}_{u'_i} + \mathbf{x}_1) \right)^{1/2} \\
&= \frac{1}{q-1} \sum_{\substack{u_i, u'_i \\ u_i \neq u'_i}} \sum_{u_1^{i-1}} \sum_{y_{21}^N} \\
&\left(P^{(2)}_N(y_{21}^N, u_i, u_1^{i-1}) \right)^{1/2} \\
&\times \left(P^{(2)}_N(y_{21}^N, u'_i, u_1^{i-1}) \right)^{1/2} \\
&= Z(U_i | Y_{21}^N, U_1^{i-1}). \quad (37)
\end{aligned}$$

Thus,

$$Z(U_i | Y_{11}^N, U_1^{i-1}) \geq Z(U_i | Y_{21}^N, U_1^{i-1}), \quad (38)$$

which had to be proved.

VII. ACKNOWLEDGEMENT

We would like to thank Prof. P. M. Olmos and Prof. T. Koch for their valuable suggestions.

REFERENCES

- [1] E. Arikan, "Channel polarization: a method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," *IEEE Transactions on Information Theory*, vol. 55, no. 9, pp. 3051–3073, Jul. 2009.
- [2] E. Sasoglu, I. Telatar, and E. Arikan, "Polarization for arbitrary discrete memoryless channels," in *Information Theory Workshop, 2009. ITW 2009. IEEE*, Oct 2009, pp. 144–148.
- [3] A. G. Sahebi and S. S. Pradhan, "Multilevel polarization of polar codes over arbitrary discrete memoryless channels," in *Communication, Control, and Computing (Allerton), 2011 49th Annual Allerton Conference on*. IEEE, 2011, pp. 1718–1725.
- [4] W.-C. Park and A. Barg, "Polar codes for q-ary channels," *Information Theory, IEEE Transactions on*, vol. 59, no. 2, pp. 955–969, 2013.
- [5] P. Trifonov, "Efficient design and decoding of polar codes," *Communications, IEEE Transactions on*, vol. 60, no. 11, pp. 3221–3227, November 2012.
- [6] M. Seidl, A. Schenk, C. Stierstorfer, and J. Huber, "Polar-coded modulation," *Communications, IEEE Transactions on*, vol. 61, no. 10, pp. 4108–4119, October 2013.
- [7] S. B. Korada, "Polar codes for channel and source coding," Ph.D. dissertation, EPFL, Lausanne, 2009.
- [8] N. Hussami, S. Korada, and R. Urbanke, "Performance of polar codes for channel and source coding," in *Information Theory, 2009. ISIT 2009. IEEE International Symposium on*, June 2009, pp. 1488–1492.
- [9] E. Arikan, "Source polarization," in *Information Theory Proceedings (ISIT), 2010 IEEE International Symposium on*, June 2010, pp. 899–903.
- [10] M. Karzand and I. Telatar, "Polar codes for q-ary source coding," in *Information Theory Proceedings (ISIT), 2010 IEEE International Symposium on*, June 2010, pp. 909–912.
- [11] E. Sasoglu, "Polarization and polar codes," *Foundations and Trends® in Communications and Information Theory*, vol. 8, no. 4, pp. 259–381, 2011. [Online]. Available: <http://dx.doi.org/10.1561/01000000041>
- [12] R. Mori and T. Tanaka, "Source and channel polarization over finite fields and Reed-Solomon matrices," *Information Theory, IEEE Transactions on*, vol. 60, no. 5, pp. 2720–2736, May 2014.
- [13] C. Hartmann and L. Rudolph, "An optimum symbol-by-symbol decoding rule for linear codes," *IEEE Transactions on Information Theory*, vol. 22, no. 5, pp. 514–517, Sep 1976.
- [14] U. Wachsmann, R. Fischer, and J. Huber, "Multilevel codes: theoretical concepts and practical design rules," *Information Theory, IEEE Transactions on*, vol. 45, no. 5, pp. 1361–1391, Jul 1999.
- [15] E. Arikan, "Coding and modulation. A polar coding viewpoint." Munich Workshop on Coding and Modulation, July 2015.
- [16] A. Bravo-Santos, "Polar codes for the Rayleigh fading channel," *IEEE Communications Letters*, vol. 17, no. 12, pp. 2352–2355, December 2013.
- [17] M. Andersson, V. Rathi, R. Thobaben, J. Kliewer, and M. Skoglund, "Nested polar codes for wiretap and relay channels," *IEEE Communications Letters*, vol. 14, no. 8, pp. 752–754, Aug. 2010.
- [18] A. Bravo-Santos, "Polar codes for gaussian degraded relay channels," *IEEE Communications Letters*, vol. 17, no. 2, pp. 365–368, February 2013.
- [19] E. Sasoglu, "Polar codes for discrete alphabets," in *Information Theory Proceedings (ISIT), 2012 IEEE International Symposium on*, July 2012, pp. 2137–2141.
- [20] S. Korada, E. Sasoglu, and R. Urbanke, "Polar codes: Characterization of exponent, bounds, and constructions," *Information Theory, IEEE Transactions on*, vol. 56, no. 12, pp. 6253–6264, Dec 2010.
- [21] H. Wymeersch, H. Steendam, and M. Moeneclaey, "Log-domain decoding of ldpc codes over GF(q)," in *Communications, 2004 IEEE International Conference on*, vol. 2, June 2004, pp. 772–776 Vol.2.
- [22] A. Lapidoth, *A Foundation in Digital Communication*. New York, NY, USA: Cambridge University Press, 2009.
- [23] R. L. Graham, B. D. Lubachevsky, K. J. Nurmela, and P. R. Östergård, "Dense packings of congruent circles in a circle," *Discrete Mathematics*, vol. 181, no. 1, pp. 139–154, 1998.
- [24] E. Spetch, "Packomania web site," 2009, [Online; accessed 22-September-2015]. [Online]. Available: <http://www.packomania.com>
- [25] G. Caire, G. Taricco, and E. Biglieri, "Bit-interleaved coded modulation," *IEEE Transactions on Information Theory*, vol. 44, no. 3, pp. 927–946, May 1998.
- [26] ETSI EN 302 755 V1.3.1 (2015-07), "Digital Video Broadcasting (DVB); Frame structure channel coding and modulation for a second generation digital terrestrial television broadcasting system (DVB-T2)," Jul 2015.